Design Migration from XC2000/ XC3000 to XC5200

XAPP 061 September 23, 1997 (Version 2.1)

Application Note by Marc Baker

Summary

The XC5200 delivers the most cost-effective solution for high-density, reprogrammable logic designs not requiring the dedicated XC4000 Select-RAM™, or very high performance of the XC3100A and XC4000 Series. This Application Note reviews the differences between the XC5200 and XC2000/XC3000 families, recommends approaches for converting XC2000/XC3000 designs to the XC5200 architecture, and provides a methodology to migrate designs easily in multiple CAE environments.

Xilinx Family

XC5200

Demonstrates

Migrating XC2000/XC3000 designs to XC5200 devices

Introduction

The new SRAM-based XC5200 FPGA family from Xilinx delivers the lowest cost-per-gate of any FPGA family. The Xilinx XC5200 family can be used in designs that originally targeted the older XC2000/XC3000 families, shown in Table 1.

The XC5200 family supports higher system clock frequencies than the XC2000 and original XC3000, but may be slower than some members of the XC3100A family.

The XC2000 family and some variations of the XC3000 family are no longer recommended for new designs. These include the XC3000 standard family, and the XC3100 family. These have been superseded by the newer XC3000A and XC3100A families. While designs in the older XC3000 families can be easily converted to the newer XC3000A families, the XC5200 family should also be considered as a lower cost alternative.

The XC5200 is not footprint compatible with the XC2000/ XC3000 families. Conversion from either the XC2000 or XC3000 to the XC5200 may require schematic and pin assignment changes. The XC5200 family does not have storage elements in the I/O, although it adds latches in the CLBs. The XC5200 offers special cascade logic for combinational functions, and carry logic for arithmetic functions, supported by new components in the library. There are many additional architectural and library differences.

Table 1: XC2000/XC3000 Families

XC2000	Original Xilinx FPGA family
XC2000L	3.3-V version of XC2000
XC3000	Second-generation FPGA family
XC3000A	Enhanced XC3000 family
XC3000L	Enhanced 3.3-V XC3000 family
XC3100	Faster XC3000 family
XC3100A	Fastest enhanced XC3000 family
XC3100L	Faster enhanced 3.3-V XC3000 family

Following the guidelines presented in this Application Note will greatly improve the chance of a successful migration.

Overview

- Design Performance

  Significant architectural differences between the XC2000/XC3000 and XC5200 make timing changes different.

  Recommendation: Use XACT-Performance™ to achieve the highest migration success. Use XDelay and simulation to verify performance prior to production. If performance is not sufficient in the fastest XC5200 device, consider migrating to the XC3100A or XC4000 Series instead.

- Three-State Buffers

  Unlike the XC3000, the XC5200 three-state buffers cannot be used to implement wired ANDs. No pullups are available to force a longline high when all buffers are disabled. The XC2000 does not have three-state buffers.

  Recommendation: Replace Wired ANDs with wide logic functions to take advantage of the cascade logic. Remove pull-ups from 3-state buffer outputs, or consider using logic gates instead. Multiplexers built from 3-state buffers may need an additional BUFT. See Figure 1 on page 6.

- Oscillator

  The XC5200 has no equivalent to the XTAL oscillator in the XC2000/XC3000, which supports an external crystal. However, the XC5200 has an internal oscillator that outputs a nominal 16 MHz frequency and many of its derivatives.

  Recommendation: For a circuit that requires a clock with an accurate frequency, use an accurate external clock.

- Input/Output Blocks (IOBs)

  Unlike the XC2000/XC3000 families, the XC5200 Input/Output Blocks (IOBs) do not contain flip-flops or latches. The XC5200 library contains macros using CLB storage elements that are identical in function to the XC2000/XC3000 primitives for IOB storage elements. Thus, no schematic changes are necessary.

  Recommendation: If you are using flip-flops or latches at the I/O, use XACT-Performance and consider locking the CLB and IOB pair to achieve faster I/O timing.

- CLB Flip-Flops and Latches

  XC2000 CLB flip-flops and latches have both asynchronous Preset (Set) and Clear (Reset). An XC5200 implementation would require that one be changed to synchronous.

- Global Clock Buffers

  There are four global buffers (BUFGs) in the XC5200 family. Each can be driven from its dedicated pads or from any internal resources. These buffers can drive to clock pins as well as control/data pins. The XC2000/XC3000 have only two clock buffers.

  Recommendation: For best performance, use the dedicated pin for the global buffer if the clock is sourced externally. Consider using these buffers for other high-fanout signals.

- Carry Logic

  The XC5200 CLBs include dedicated carry logic for a fast carry capability. This improves the performance and density of arithmetic functions like adders, subtractors, comparators, accumulators, and counters. Dedicated carry is not available in the XC2000/XC3000 families.

  Recommendation: Use the macros in the Unified Library that take advantage of carry logic. Change counters to the macros that use the prefix “CC” to use the carry.

- Cascade Logic

  The XC5200 carry logic may also be used to implement wide logic functions.

  Recommendation: Replace cascaded four- or five-input gates with the eight-to-sixteen-input gates in the XC5200 library, to use cascade logic.

Design Guidelines & Considerations

Because the XC5200 has a different architecture (see Table 3), there are a number of issues to consider before migrating an XC2000/XC3000 design to the XC5200.

Configuration

The XC5200 configuration process is more similar to the XC4000 than to the XC2000/XC3000. It also supports two additional modes: Synchronous Peripheral and an Express mode that shortens configuration time. The Express mode is similar to the slave mode but instead of transferring a single bit of data per cycle, it transfers a byte of data per cycle. Taking advantage of Express mode reduces configuration time by a factor of eight.

Footprint

XC2000/XC3000 and XC5200 devices are not footprint compatible. The XC5200 is only footprint compatible with the XC4000 and Spartan families. However, most XC2000/XC3000 packages are available for the XC5200, and the VersaRing™ I/O routing of the XC5200 helps maintain pinouts after design changes.

Density

The XC2000 family has two members, the XC2064 and XC2018. Even the larger XC2018 should fit in the smallest XC5200 device, the XC5202 (see Table 2). The XC3000 families should fit in the mid-range members of the XC5200 family.

Table 2: Density Comparison

XC2000/ XC3000	Max Gates	XC5200	Max Gates
XC2064	1,000	XC5202	3,000
XC2018	1,500	XC5202	3,000
XC3020	1,500	XC5202	3,000
XC3030	2,000	XC5202	3,000
XC3042	3,000	XC5204	6,000
XC3064	4,500	XC5204	6,000
XC3090	6,000	XC5206	10,000
XC3195	7,500	XC5206	10,000

Table 3: Comparison of Resources Between XC5202, XC5206, XC5210 and XC3195, XC2018

Resource	XC5202	XC5206	XC5210	XC3195	XC2018
Max Logic Gates	3,000	10,000	16,000	7,500	1,500
Maximum CLB flip-flops/latches	256	784	1,296	968/0	100
Maximum IOB flip-flops/latches	0	0	0	352/176	74/0
Maximum I/O pins	84	148	192	176	74
Configuration bits	42,416	106,288	165,488	94,984	17,878
Function generators	256	784	1,296	968	100
Flip-flops per CLB	4			2	2
Global buffers	4			2	2
Cascade capability	Yes			No	No
Carry logic	Yes			No	No
Internal 3-state drivers per horizontal line	10	16	20	23	0
	no pullups			pullups	
Output drive	8 mA			4 mA	4 mA
Output slew rate control	Yes			Yes	No
Boundary-scan	Yes			No	No
Internal oscillator	Yes			XTAL	XTAL
Configuration modes	7			5	5
Packages	PC44, VQ64, PC84, PQ100, VQ100, TQ144, PG156	PC84, PQ100, VQ100, PQ160, TQ144, TQ176, PG191, PQ208	PC84, PQ160, TQ144, TQ176, PG223, PQ208, PQ240, BG225	PC84, PQ160, PG175, PG223, PQ208	PC44, PC68, PC84, VQ64, TQ100, PG84

Design Performance

Because the XC5200 was engineered primarily for low cost, the fastest XC5200 device may be slower than the fastest XC3100A in some applications. However, there are many significant new features in the XC5200 such as Boundary Scan, Carry Logic, Startup, Internal Oscillator, VersaBlock™, VersaRing, and Express Configuration. In some cases, these features enable the XC5200 to out-perform XC2000/XC3000 devices in a similar speed grade.

The XC2000 and original XC3000 families use speed designators in the part number based on maximum frequency, not CLB delay as in the XC5200 family. See Table 4 for a cross-reference. Note that the XC5200 offers speed grades faster than the XC2000/L, XC3000/A/L, and XC3100L families, and comparable speed grades to the XC3100A family.

Use XDelay and simulation to verify performance prior to production. XDelay can be used to report performance at various speed grades without changing the LCA file. To show the delays of the most critical paths, create a short XDelay report using the following command:

xdelay -u <speed> -o critical.rpt <design_name>

For example:

xdelay -u -3 -o critical.rpt new.lca

This command produces a text file called critical.rpt that contains the minimum worst-case pad-to-setup, clock-to-setup, and clock-to-pad values allowable for each clock in the design. Effectively, this report provides all the information necessary to evaluate the performance of the new speed grade.

Alternatively, in the Windows environment use the Performance Summary in the Timing Analyzer. Both XDelay and the Timing Analyzer can also be used to examine specific path delays in more detail if desired.

Table 4: XC2000/XC3000 Speed Designators vs. XC5200

XC2000		XC3000		XC5200	
Part	t_{ILO} (ns)	Part	t_{ILO} (ns)*	Part	t_{ILO} (ns)
L-10	9.5	L-8	6.7/7.5		
-70	10	-70	9		
-100	8	-100	7		
-130	5.5	A-7	5.1/5.6	-6	5.6
		-125	5.5		
		A-6	4.1/4.6	-5	4.6
		-5	4.1/4.6		
		-4	3.3/3.72	-4	3.8
		-3	2.7/3.05	-3	3
		-2 (L-2)	2.2/2.55 (2.49)		
		-1	1.75/2.05		
		-09	1.5/1.8		

* FG mode/F and FGM mode

Because the XC5200 has a different architecture, most of the XC2000/XC3000 mapping, placement, and routing information will be irrelevant when migrating to the new XC5200. The PPR Guide option cannot be used. Consequently, the best way to ensure that design performance will be met in the XC5200 is to use XACT-Performance to define the timing requirements of the design.

Development System

The XC5200 is supported by the same XACTstep™ development system and uses the same basic software tools that support the XC2000/XC3000. Newer place & route algorithms are designed specifically for the XC5200 architecture. The XC5200 family is supported by all the latest Xilinx entry, implementation, and verification tools. User constraints files generated for earlier versions of the place and route tools may need to be updated.

A new XC5200 library has been added to the set of Unified Libraries to support the architectural features of the XC5200. These new library elements are described in detail in the *Libraries Supplement Guide*.

All Unified Library elements for the XC2000/XC3000, with the exception of those listed in Table 5, are compatible with the XC5200. Library differences result from the XC5200 family's lack of Asynchronous Preset and oscillator amplifier.

Table 5: XC2000/XC3000 Unified Library Symbols Not Compatible With XC5200

XC2000/XC3000	XC2000 only
CLB	ACC1
IOB	ADD1
CLBMAP	ADSU1
GXTL	FDCP(E)
OSC	FJKCP(E)
GCLK => BUFG	FTCP(E)
ACLK => BUFG	FTCPLE
	LDCP(E)

Many new XC5200 optimized elements were added to the XC5200 schematic library. These elements are described in the following architectural descriptions.

Carry Logic

The XC5200 family has dedicated carry logic to ripple the carry of all four bits within a single CLB. This carry logic can be cascaded to adjacent CLBs above to form a larger chain. All adders, subtractors, incrementers, decrementers, accumulators and some counters in the XC5200 library are implemented with carry logic, using the CY_MUX symbol, and Relationally Placed Macros (RPMs). The counters implemented with carry logic begin with "CC". The RPMs can also be used as examples when implementing customized RPMs.

Cascade Logic

The XC5200 carry logic also offers a very efficient and high speed cascade capability. A good example is the implementation of a wide decoder function. Each logic cell can decode four bits. With the cascade ability, four logic cells within a single CLB can form a 16-bit decoder. Wider decoders can be implemented by cascading more CLBs. This is a better implementation of address decoding since the area and performance cost for cascading is much less than multiple levels of CLBs. This cascade capability may also be used to optimize comparators, parity generators, or any other wide homogeneous function.

Cascade logic is automatically used in gate functions of eight inputs or more. The XC5200 library adds 12-input and 16-input gates, as well. The DECODE macros in the XC5200 library use the same cascade logic as in the wide AND gates. The XC5200 wide gates should also be considered as replacements for cascaded gates in the XC2000/XC3000.

The XC5200 adds a new library component called F5_MUX to create any five-input function in one CLB level. The F5_MUX is a 2-to-1 lookup table multiplexer primitive. It allows two adjacent lookup tables in an XC5200 CLB to be combined together into one result without using another lookup table. The F5_MUX is not used automatically, as five-input functions can be effectively implemented in two levels of logic. The F5_MAP mapping symbol forces the use of the F5_MUX. Add the F5_MAP to any five-input functions that demand the highest possible speed.

Three-State Buffers

The XC5200 offers internal three-state buffers (BUFTs), similar to the XC3000; the XC2000 does not have three-state buffers. The XC5200 has four BUFTs per logic block, each sharing a common output enable (OE) line. Each BUFT can be driven by any one of the CLB outputs, and can drive two of the eight horizontal or vertical longlines. The data inputs to the BUFTs are more restrictive than the XC3000 in that they can only be driven by the outputs of the adjacent CLB.

These differences should not be an issue when migrating designs from XC3000 to XC5200, but one additional difference could make an impact. Unlike the XC3000, the XC5200 has no pullups on the ends of the long lines sourced by BUFTs. Consequently, wired functions and wide multiplexing functions requiring pullups in undefined states (i.e., some 3-state bus applications) cannot be implemented directly in the XC5200 architecture. Note that in the absence of a pullup, the weak keeper circuit avoids undefined logic levels in both XC3000 and XC5200 families.

There are some simple workarounds for these restrictions. In the case of the wired functions, the same functionality can be achieved by taking advantage of the carry/cascade logic described above, implementing a wide logic function in place of the wired function. In the case of the 3-state bus applications, if the undefined states are “don’t cares,” then the XC5200 weak keeper circuit will retain the previous value on the bus lines when all drivers are 3-stated, and no action is required. If the undefined states are required to be pulled up, then the user must make sure that all states of the multiplexing function are defined. This process is as simple as adding an additional BUFT to drive the bus High during any of the previously undefined states. See Figure 1 on page 6 for an example.

Input/Output Pads

The I/O blocks in the XC5200 differ from those in the XC2000/XC3000 in that they contain no flip-flops or latches. An XC2000/XC3000 IOB flip-flop/latch component will automatically convert to an XC5200 CLB flip-flop/latch macro. However, you may want to consider taking advantage of the lookup table or control signals available to a CLB flip-flop/latch. See the *Libraries Supplement Guide* for a complete description of the variations on the XC5200 latch library components.

In order to achieve performance comparable to an XC2000/XC3000 input flip-flop (IFD), the XC5200 input pad must drive one of the flip-flops in the adjacent CLB. Similarly, an XC3000 input latch (ILD) can be emulated in an XC5200 by driving an LD in the CLB from an adjacent pad. This important placement and routing combination can be achieved by using XACT-Performance to indicate to PPR the timing requirements on the pad-to-setup path, or by using the XACT-Floorplanner.

XC5200 IOBs default to a slow slew rate to reduce ground bounce, as in the XC3000 IOBs. Select a fast slew rate where speed is critical. The XC2000 does not have this option.

XC5200 IOBs have 8 mA output drive, equivalent to the XC3100/A families and higher than the XC2000/XC3000/A families (4 mA).

IOB/CLB Logic

IOB and CLB logic symbols are not supported in the XC5200 library. To get an equivalent function, the user will need to refer to the reference manual on the following topics: FMAP, F5MAP, LOC=, RLOC=, and HBLKNM=.

Flip-Flops and Latches

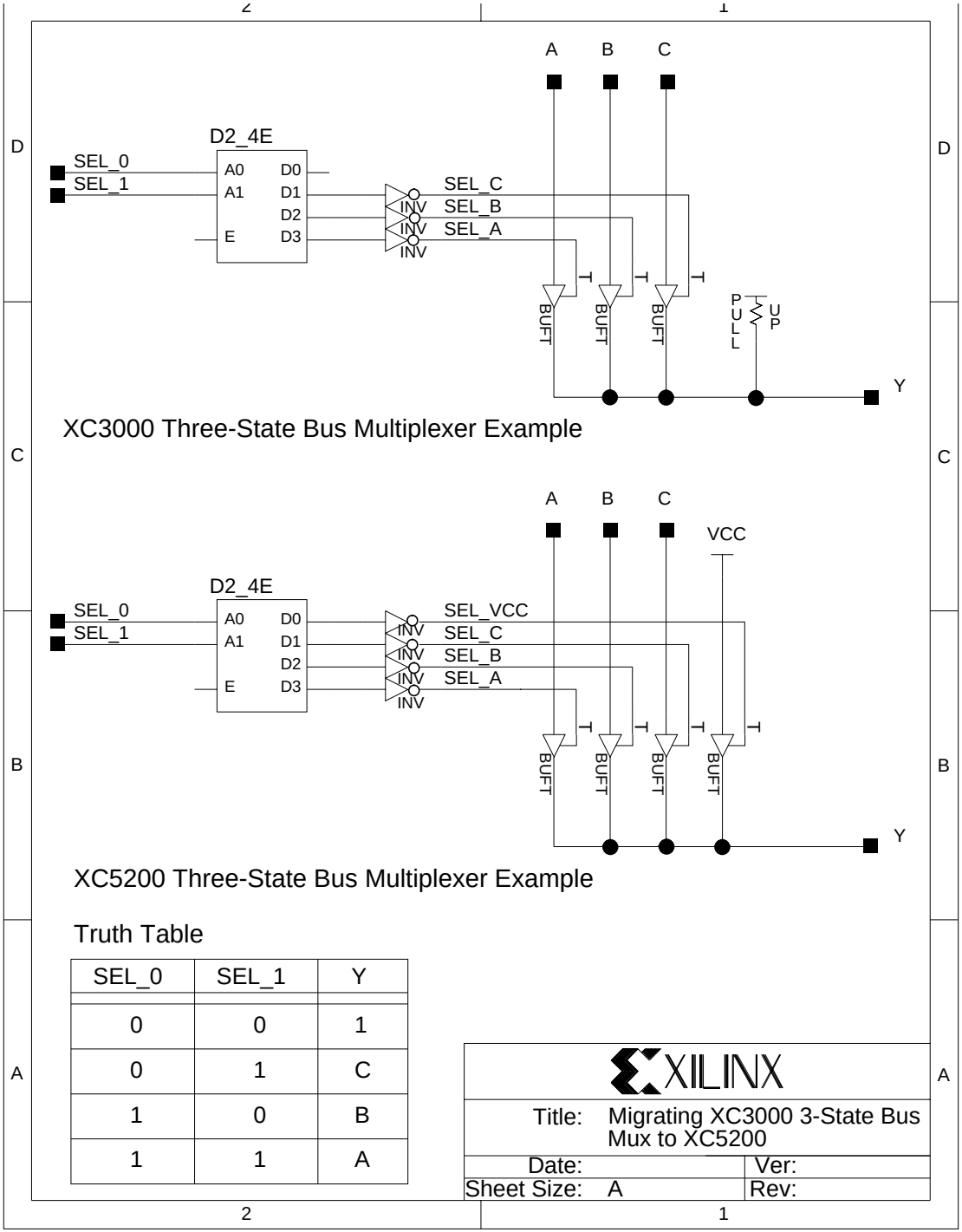
Both the XC5200 and the XC2000/XC3000 provide flip-flops in the CLBs. The XC5200 and XC2000 also provide level-sensitive latches.

Both flip-flops and latches have an Asychronous Clear (Reset) in the XC5200, XC3000, and XC2000. The XC2000 adds an Asynchronous Preset (Set) input to each storage element, available at the same time as the Clear. If both asynchronous Preset and Clear are being used in the XC2000, one would have to be eliminated, or converted to synchronous, in the XC5200. To emulate an Asynchronous Preset only, add an inverter to the D and Q pins of the flip-flop or latch.

The XC5200 offers clock enables for all flip-flops and latches, which are not available in the XC2000 or in the XC3000 IOBs. Use clock enables instead of gating clocks.

Global Clock Buffers

The XC5200 has more global buffers (four) than the XC2000/XC3000 (two). Global buffers in Xilinx FPGAs are special buffers that drive a dedicated interconnect network throughout the device. This network is specialized for the distribution of high fanout clocks or other control signals, such that it minimizes delay and skew while distributing the signal to many loads. Additional delay will be added to the distribution of the clock signal if the dedicated pads for the global buffers are not used.



Truth Table

SEL_0	SEL_1	Y
0	0	1
0	1	C
1	0	B
1	1	A

XILINX		
Title:	Migrating XC3000 3-State Bus Mux to XC5200	
Date:		Ver:
Sheet Size:	A	Rev:

Figure 1: Multiplexing in 3-State Bus Applications

The XC5200 library uses the component BUFG to represent the global buffers. The XC2000/XC3000 libraries use the components GCLK and ACLK.

Oscillator

There is no XC5200 equivalent to the XC2000/XC3000 XTAL amplifier for an external crystal. The XC5200 internal oscillator is very different. Use the internal oscillator with consideration that the frequency variations may be up to 50%. Any circuit that uses an external crystal can no longer be implemented - try using a precise external oscillator.

The XC5200 oscillator lets the user choose to divide either the internal 16 MHz clock, using the OSC5 component, or a user clock which is connected to the 'C' pin of the CK_DIV component. The user can use both the 'OSC1' and 'OSC2' outputs of the symbol, and has the choice of division by 4, 16, 64, or 256 on pin 'OSC1' and a division by 2, 8, 32, 128, 1024, 4096, 16384, or 65536 on pin 'OSC2'. The division in each case is user-specified by adding an attribute to the symbol: DIVIDE1_BY= for 'OSC1' or DIVIDE2_BY= for 'OSC2'.

Boundary Scan

The XC5200 family adds boundary scan, which is not available in the XC2000/XC3000 families. The XC5200 supports all the mandatory boundary-scan instructions specified in the IEEE standard 1149.1 and more. The library symbol for boundary scan is called BSCAN.

Global Reset

The XC2000/XC3000 families offer a dedicated active-low global reset pin. If the current XC2000/XC3000 design calls for the use of global reset or global 3-state, the XC5200 offers a better solution, although it requires an addition to the schematic. Use the STARTUP symbol from the XC5200 library (see Figure 2). Activation of the GR signal asynchronously resets all flip-flops and latches in the design. Note that it is an active-high reset, unlike the XC2000/XC3000 Reset pin. To put all I/Os into 3-state mode, drive a logic 1 onto the GTS (global three-state) pin of the symbol. Both GR and GTS are invertible and can be driven by any source within the device as well as any IOB.

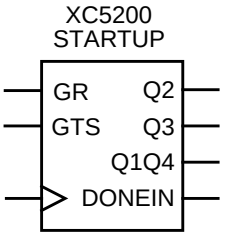


Figure 2: XC5200 STARTUP

If an asynchronous preset is desired in an XC5200 design, inverters can be added to the D and Q pins of the flip-flop to create equivalent functionality.

Configuration

The XC5200 devices use a different configuration process than the XC2000/XC3000. However, XC5200 devices may be used in daisy chains with XC2000/XC3000 devices. The XC5200 $\overline{\text{PROGRAM}}$ pin is a single-function input pin that overrides all other inputs. It can be used to hold off or abort configuration, as RESET is used on the XC2000/XC3000. The DONE pin is separate, unlike the XC2000/XC3000. The XC5200 $\overline{\text{INIT}}$ pin also acts as a Configuration Error output.

XC5200 devices support two additional programming modes: Peripheral Synchronous and the new high-speed Express mode. The peripheral modes in the XC3000/XC5200 load in parallel, while the XC2000 peripheral mode is serial. Readback in the XC5200 family either ignores the flip-flop content, thereby avoiding the need for masking, or it takes a snapshot of all flip-flops at the start of Readback, avoiding the need to stop the system clock. Readback in the XC5200 has the same polarity as Configuration, and can be aborted.

Configuration Startup

Start-up is the transition from the configuration process to the intended user operation. The XC5200 offers a programmable superset of the capabilities of the XC2000/XC3000 families (see Figure 3).

The XC2000 family goes through a fixed sequence. DONE goes High and the internal global Reset is deactivated one CCLK period after the I/O become active.

The XC3000 family can be programmed to have DONE go High one CCLK period before or after the I/O become active. Independent of DONE, the internal global Reset can be programmed to be deactivated one CCLK period before or after the I/O become active.

The XC5200 family offers additional flexibility. The three events - DONE going High, the internal global Reset being de-activated, and the user I/O going active - can all occur one CCLK period before or after, or simultaneous with, the others. The default option, and the most practical one, is for DONE to go High first, disconnecting the configuration data source and avoiding any contention when the I/Os become active one clock later. Reset is then released one period later to make sure that user operation starts from stable internal conditions. The designer can modify the sequence to meet particular requirements by means of software options in the bitstream generator.

XC5200 start-up can be synchronized to any user clock by means of a configuration option.

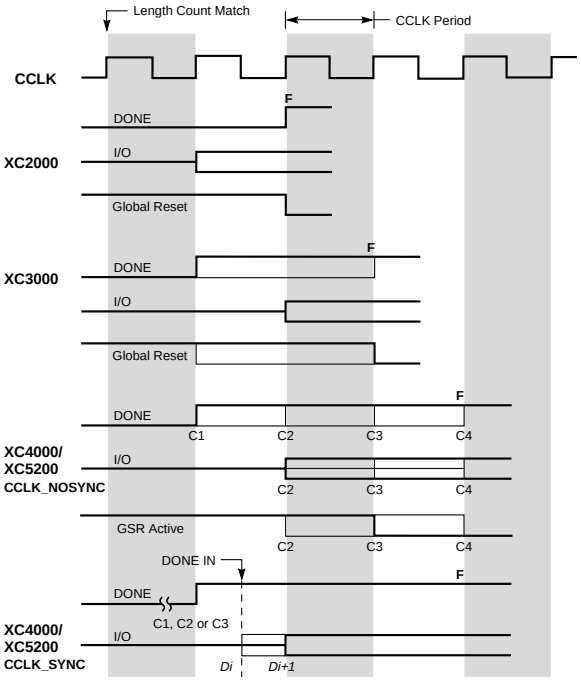


Figure 3: Start-Up Timing

Power-Down

The XC5200 has no equivalent to the built-in powerdown logic of the XC2000/XC3000 families, and has higher standby power. The powerdown logic, when activated via the PWRDWN pin, will disable normal operation and retain only the configuration data, even if VCC is lowered to 2.3V. The XC2000 families and XC3000/A families can be powered down to a current consumption of a few microamps. These families consume less than 1 mA at standby, even if not in powerdown mode. The XC3100/A still draws 5 mA, even in power-down. The XC5200 family draws 15 mA max at standby, and VCC must stay above 4.5V to prevent reconfiguration. Selecting CMOS input thresholds (available in all families) reduces supply current considerably, as does shutting down external clock and logic activity. The XC5200 offers a global three-state net that does not reset any flip-flops.

Migration Methodology

Using the Unified Libraries, it is possible to migrate an XC2000/XC3000 design to the XC5200 architecture, and thus take advantage of significant cost reductions. The migration methodology itself is simple, and following the guidelines and considerations prescribed in this document will greatly improve the success of the migration.

This section describes how to perform the actual migration of an XC2000/XC3000 design into the XC5200 architecture for three third-party CAE interfaces. The XC3000 is used as an example.

VIEWlogic

To migrate an XC2000/XC3000 VIEWlogic schematic to the XC5200 architecture, perform the following steps:

1. Add the XC5200 library to the VIEWlogic library search path. Edit the **`viewdraw.ini`** file in the project directory and add the XC5200 library path so that it appears in **`viewdraw.ini`** before the path to the XC3000 library.
2. To convert the XC3000 alias to XC5200, run altran, the VIEWlogic library alias maintenance program:

   `altran -l primary xc3000=xc5200`
   where **`xc4000`** is the alias assigned to the XC3000 library; and **`xc5200`** is the alias assigned to the XC5200 library.

3. Reprocess the design by running XMake or the Flow Engine.

Mentor

To migrate an XC2000/XC3000 Mentor schematic to the XC5200 architecture, perform the following steps:

1. Invoke **PLD_DA** (it is not necessary to open the schematic).
2. On DA's desktop background (that is, outside of any schematic or symbol windows), call up the session pop-up menu with the mouse button on the right and select **Convert Design**.

   Of the fields in the resulting dialog box these are the most relevant:

3. **Select a group of designs from a list file?** Whether you answer "**yes**" or "**no**" to this question affects the following field.
4. **Enter Design name (List file = no)**. The name of the design to retarget. **Convert Design** does *not* traverse the hierarchy of a schematic.
5. **Enter list file name (List file = yes)**. A file which lists designs, one per line, to retarget. This is useful if your design has many lower-level schematics.

TIP: A list file can easily be created by typing:

```
ls *.mgc_component.attr | sed
 s/.mgc_component.attr//g > listfile
```

The **ls** command lists all MGC components within a single directory. The **sed** command strips away the **.mgc_component.attr** trailer. The result is redirected to **listfile**.

6. **Schematic name**. The name of the schematic model (the default is "**schematic**").
7. **Check & Save switch**. Because all schematic sheets in **Convert Design** are literally re-drawn in Design Architect, you must apply **Check & Save** to each sheet. This switch controls whether to do this automatically. By default, this switch is set for manual checking because it allows you to spot Xilinx components that did not convert properly. Use the manual setting until you are comfortable with how **Convert Design** works and you are certain that all Xilinx components will convert properly.
8. **From Technology**. The device family from which you are converting (e.g., **XC3000**). This and the next field are case insensitive.
9. **To Technology**. The device family to which you are converting.
10. After filling out the fields in the dialog box and selecting "**OK**," you will see Convert Design doing its job directly in Design Architect.
11. Reprocess the design with XMake or the Flow Engine.

Foundation

To migrate an XC2000/XC3000 Foundation schematic to the XC5200 architecture, perform the following steps:

1. Select the **Project Type** option from the **Menu** file. Change **Family**, **Part**, and **Speed** settings, as desired, and click the **Change** button.
2. Open and save each schematic sheet macro. To do so, run the Schematic Editor and select the **Open** option from the **File** menu. The Open Sheet window allows you to quickly open all project top-level sheets and project schematic macros. Inspect the Project Manager messages for any warnings and errors.
3. Re-synthesize and update all FSM and HDL macros. Use the hierarchy browser in the Project Manager to search the project for the macros.
4. Note: if your project contains components that are not available in the new system library, you have to modify the project so as to preserve its functionality. In the case of a top level HDL project, you will need to re-synthesize the entire project.
5. Reprocess the design with XMake or the Flow Engine.

Additional Information

If there are problems with the conversion process, please contact the Xilinx Technical Support hotline for assistance.

Email: hotline@xilinx.com (24 hours)

Voice: 1-800-255-7778 (6:30AM - 5:00PM PST)

FAX: 1-408-879-4442 (24 hours)

Web: www.xilinx.com (click on Answers)